UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, STATION A, MONTRÉAL, QUÉBEC, H3C 3J7

PUBLIC TRANSIT NETWORK ASSIGNMENT METHOD

by

Claude Chriqui

To be presented at the Summer Annual Meeting of Planning & Transport Research & Computation (PTRC), London, July 1975.

Université de Montréal
Rapport de recherche # 75-08 - Ecole des Hautes Etudes Commerciales
Publication # 21 - Centre de recherche sur les transports
Mai 1975

Dépôt légal - 1er trimestre 1978

1. INTRODUCTION

Most of the assignment procedures in public transit network use the traditional "all or nothing" method applied on a computer model of the network. Because the problem is very close to the road assignment problem, the arc-node representation has been widely used. So does Dial [3]. In that case he disregards the fact that an arc may be served by several distinct lines as he creates a "super arc" (the "trunk-line link"). In this method one must, for each two adjacent nodes, compute the expected travel time. This can be a long task. Dial does not take advantage of the presence of the routes. Le Clercq [7] proposed a line-node representation of the network and defined the distance between two adjacent nodes as the time taken by the fastest route to join them. He states the problem of common bus lines and recognizes that it is as a combinational problem but leaves it unsolved.

Dealing with large networks one is faced with this problem of describing the assignment on each existing line between two nodes. These lines may be served by metros, buses, express routes at peak hours, etc... Based on a line-node structure for the network, the method presented here can handle Le Clercq's above definition as well as the common lines case.

2. STRUCTURE

The network representation we used is fully described by Chapleau [1]. We outline it here. The required input data consist of (1) for each node a triplet of numbers: its identifier and its x - y coordinates in a geocoded system and (2) for each route its identifier, frequency, commercial velocity and a list of identifiers for nodes. Routes are defined for use in both directions and provision is made for different paths on these directions. The output of the procedure consists of a computer file. By the use of pointers one can trace nodes on any line and describe all the lines passing through any node. Travel time between two adjacent nodes on a given route is also available in the file.

3. THE ASSIGNMENT METHOD

We will build a minimum travel time tree where the travel time between two nodes A and B is defined by an impedance function $S(A, B, \mathcal{L})$, where $\mathcal{L}$ is the set of lines passing through A and B. In short it is an all or nothing procedure with diversion among common bus lines.

3.1 The impedance function

One of the steps in the algorithm consists in the computation of this value. We have chosen to use two criteria but one can easily select others as well. In one case we define $S(A, B, \mathcal{L})$ as the travel time (waiting time + in-vehicle time) performed by the line $\ell \in \mathcal{L}$ for which it is minimum i.e. $S(A, B, \mathcal{L}) = T$ where $T \leq T_i \ \forall_i$ and T_i is the total travel time by using line $i \in \mathcal{L}$. This is the method used by Le Clercq. We will show later how this method may be improved upon. In the other case we define $S(A, B, \mathcal{L})$ as the "perceived travel time" in the following way .

$$S(A, B, \mathcal{L}) = W_L + D_L$$

where L is the subset of $\mathcal{L}$ for which S is optimal i.e.

$$S(A, B, L) \leq S(A, B, \mathcal{F}) \quad \forall \mathcal{F} \subseteq \mathcal{L}$$

W_L and D_L are respectively the expected waiting time and the in-vehicule time if one uses any one of the lines in L.

In short this definition states that a user faced with the choice among several lines (all taking him to his destination) will select a subset of lines which "optimizes" the expected travel time and hence the user will take the first vehicle serving any one of the selected lines to perform his voyage.

A solution to the problem of selecting L and computing S has been described by Chriqui and Robillard [2]. In [2] the general case as well as exponential and uniform distribution for headways have been developed. We outline here the method using the hypothesis of exponential distribution of the headways.

Let $\ell_1, \ell_2 \ldots, \ell_n$ the lines in $\mathcal{L}$, (Fig. 1), joining A to B,

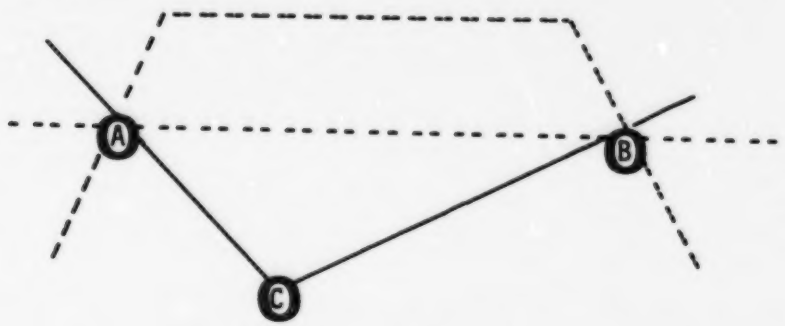

Fig. 1

$w_1, w_2, \ldots, w_n$ their waiting times and $t_1, t_2, \ldots, t_n$ the corresponding in-vehicule time.

Step 1. Sort the ℓ_i's in order of increasing t_i's i.e. now

$$\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_n\}$$

$$\downarrow \quad \downarrow \qquad \downarrow$$

$$t_1 \quad t_2 \ldots . \quad t_n$$

with $$t_1 \leq t_2 \leq \cdots \leq t_n$$

and set $T_L = t_1 + w_1 \quad ; \quad L = \ell_1$

Step 2. For $i = 2, \ldots n$ do

$$\text{if } t_i \leq T_L \text{ then } \begin{cases} T_L = \dfrac{1 + \sum_{j=1}^{i} t_j * 1/w_j}{\sum_{j=1}^{i} 1/w_j} \\ L = L \cup \ell_i \end{cases}$$

else stop

We present now the assignment algorithm where one may use either of the two behavior models or any other one may define.

Let σ be the origin. We will build a tree as done in the Dijsktra [4] method. Nodes will be given labels of the form.

$$[\, T(\sigma,A),\ k,\ L(k,A)]$$

where $T(\sigma,A)$ is the tentative minimum total travel time between σ and node A, k is the node at which took place the last transfer in the voyage between σ and A; $L(k,A)$ is the optimal subset of $\mathcal{L}(k,A)$ itself the complete set of lines passing through σ and A. Let V be the set of nodes. $H(k)$ the set of nodes $i \in V$

directly connected (without any transfer) to k, $B(k)$ the set of lines passing through k.

3.2 Algorithm

Step 1. σ is given the label $[0, \sigma, \phi]$ and all the other nodes receive the label $[\infty, 0, \phi]$

$X = \phi$ is the set of nodes labelled permanently.

Step 2. Determine k such that

$$T(\sigma, k) = \min \{T(\sigma, j),\ j \in \bar{X}\}$$

Step 3. Do $\forall\ j \in H(k)$: $\mathcal{L}(k, j) = \phi$

Step 4. Do $\forall\ \ell \in B(k)$

Do $\forall\ j \in \ell$: $\mathcal{L}(k, j) \leftarrow \mathcal{L}(k, j) \cup \ell$

Step 5. Do $\forall\ j \in H(k)$

if $T(\sigma, k) + S(k, j, L(k, j)) < T(\sigma, j)$

then j is given the label

$[T(\sigma, k) + S(k, j, L),\ k,\ L(k, j)]$

Step 6.

6.1 $X \leftarrow X \cup \{ k \}$ and the label of k becomes permanent

6.2 if $\bar{X} = \phi$ stop else go to step 2.

4. LE CLERCQ'S SPECIAL CASE

If we define S (k, j, L(k,j)) as the time taken by the fastest route in $\mathcal{L}(k,j)$ then L (k,j) contains only one line. The two following propositions improve the procedure proposed by Le Clercq.

Proposition 1

In step 2 of the algorithm if B(k) contains only one line (except for $k = \sigma$) go to step 6.

Proposition 2

In step 4 of the algorithm if $\ell = \ell_k$ (ℓ_k is the line which "takes" to k, i.e. the line indentified in the current label of k) then go to step 6.

These two propositions whose proof is immediate state that it is not necessary to scan the nodes on a specific line for no

improvement will be achieved.

As most of the nodes are traversed by two lines these propositions enables us to save a large amount of scanning time.

5. IMPLEMENTATION

As one may expect most of the computer time is spent on the step 2 when determining min $\{T(\sigma,j),\ j \in \bar{X}\}$. In order to avoid direct search for that minimum we used a binary tree structure to keep a relative sorted file of node identifiers and their relative $T(\sigma,j)$. The method is fully described by Gilsinn and Witzgall [5] and Knuth [6]. Two parallel "heaps" are used (one for identifiers, the others for the $T(\sigma,j)$ values). The top of the first heap contains the current node j such that $T(\sigma,j)$ is the minimum. Each element i of the heap is such that

$$[\,T(\sigma, m_1)]_i \;\leq\; [\,T(\sigma,m_2)]_{i+2}$$

and

$$[\,T(\sigma, m_1)]_i \;\leq\; [\,T(\sigma,m_3)]_{i+1}$$

where m_1, m_2, m_3 three distinct nodes and $[T(\sigma,m_1)]_i$ means the contents of element i of the heap. As we will see this technique permitted a great saving in computer time. The methods presented

here have been implemented on the MUCTC - Montreal Urban Community Transportation Commission - network. The system consists of 3 metro lines and 115 bus lines - 600 nodes have then been defined on this network. The computer time using the Le Clercq criterion was 0.5 seconds to find the minimum path tree on a CDC 7400 - 2.0 seconds when the heap procedures is not used! - for one origin. Performing the common bus lines algorithm took 2.5 seconds for the same network.

As one can see the two definitions described above may give raise to two algorithms. Common bus lines can be used for detailed assignment to the level of each particular route thus achieving a diversion on routes sharing common line sections. The Le Clercq alternative is fast but does not provide us with this detailed assignment. The latter can be used for a first trial assignment when one is interested in a macroscopic image of the line loading.

6. REFERENCES

[1] CHAPLEAU, R. "Réseau de transports en Commun: la structure informatique et l'affectation" Département d'Informatique, 1974, Université de Montréal.

[2] CHRIQUI, C. and ROBILLARD, P. "Common Bus lines" Publication no 180, Département d'Informatique - Université de Montréal. Accepted for publication in Transportation Science.

[3] DIAL, R.B., WOORHEES & Ass, Inc. "Transit Pathfinder Algorithm" Highway Research Board, Record no 205, 1967-67-85.

[4] DIJKSTRA, E.W., "A Note on Two Problems in connection with Graphs", Numerishe Mathematik 1, 1959

[5] GILSINN, J., & WITZGALL, C., "A Performance Comparaison of Labelling Algorithms for calculating Shortest Path Trees", National Bureau of Standards, Note 772, May 1973. U.S. Department of Commerce.

[6] KNUTH, D.E., "The Art of Computer Programming" vol. 3, Addison-Wesley, 1969.

[7] LE CLERCQ, F., "A Public Transport Assignment Method" Traffic and Engineering & Control, June 1972.